

IMAGE DATA ORGANIZATION INTO PIXEL TILE MEMORY MATRIX

Fred J. Reuter

TECHNICAL FIELD OF THE INVENTION

The technical field of this invention provides a method of manipulating and processing display element data for scanned printer image buffers.

5

BACKGROUND OF THE INVENTION

Printer page description languages (PDL), such as Postscript, use opaque image build up techniques to create the print page image. As new subimages are added to the image, the new subimage is written over the previous image within the

10

boundary of the new subimage. These subimages are two dimensional regions which are mapped into memory space and stored until the image creation is complete. This requires an image memory which is either addressable on display element boundaries or a memory which can be read, modified, and rewritten. The former requires image processors with narrow data bus widths which are not conducive to high speed data transfers. The later allows for high speed transfers but requires transfer of data which may not need to be modified.

These images consist of relatively few bits per display element but high performance processors necessary to process this type of image typically have data busses with widths which are several times wider than the number of bits in a display element.

SUMMARY OF THE INVENTION

This invention is a technique of image data processing.

Image data is stored in a memory having data words of a predetermined data width. Each data word includes a plural adjacently disposed image pixels of a single scan line. A set of consecutive data words corresponds to a two dimensional tile of the image whereby adjacent data words store image pixels of adjacent scan lines. The image data is transferred to a cache in these tiles. Following image processing on a tile of image data stored in the cache, the tile of image data is transferred back to the memory. The technique repeats for each tile of image data. Separate tiles of image data may be operated on by different data processors simultaneously.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects of this invention are illustrated in the drawings, in which:

Figure 1 illustrates the image data organization in
5 memory of this invention;

Figure 2 illustrates in block diagram form an image data processor implementing this invention; and

Figure 3 shows a block diagram of the TMS320C82 DSP in an image data processing system according to this invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The problem addressed by this invention is how to organize the image memory for fast and efficient transfer of image data from the processor to the image storage memory for
15 read, modify, write applications. This invention uses a processor with a wide data bus which can cache several words of data and organize the image memory in square tiles of display elements. This processor can cache small tiles of image memory, perform the intensive bit manipulations
20 necessary and store the tile of display elements back to the image memory.

Assume the following processor attributes in an example describing the invention. The processor data bus width is 64 bits. The processor is byte addressable, capable of
25 addressing data elements of a size of 8 bits. The display element size is 4 bits. The pixel tile size is 16 by 16 display elements.

Figure 1 illustrates the image data organization in memory of this invention. For efficiency of memory space,
30 display element data is packed into memory as 16 pixels per

long word of 64 data bits. The memory is organized with 16 long words per tile starting on a modulo 128 address boundaries in the image display memory. The 64 bits in the first long word 101 in Tile 0 represent 16 adjacent pixels.

5 The following long word 102 represents 16 pixels in the next cross process line of pixels directly below the pixels in the first long word. This sequence continues until 16 long words of pixel data has been defined ending with the 16 pixels of the sixteenth long word 116. The seventeenth long word 117,
10 the first long word of the next tile, Tile 1, represents the 16 pixels adjacent in the cross process direction from the first long word in the last tile. This sequence continues until the far side of the image is included, then the sequence of tiles restarts 16 rows below the previous sequence of
15 tiles. Note in Figure 1, the numbers within the boxes are the offset byte addresses from the beginning of the image in Hexidecimal.

Prior systems use processors without data caches. These processors must utilize the data bus for the entire read,
20 modify, write cycle for every display element manipulation.

These prior systems organized the memory as one-dimensional arrays of pixels, thus requiring additional accesses to perform associative operations in the second dimension.

This invention enables the processor to make relatively
25 few memory bus accesses, in this example 16, in order to load a two dimensional array of display elements. This array can be operated upon from within the processor's cache and then returned to the image memory with only a few additional memory bus accesses. This reduces the time and overhead associated
30 with accessing the image memory bus for each operation on each

pixel element.

This solution reduces the amount of image memory bus activity associated with display element processing allowing more processors to have access to the image memory to operate on different areas of the image memory at the same time. This will enable higher performance display processing without the need to increase memory speed or memory bus bandwidth.

Figure 2 illustrates in block diagram form an image data processor 200 implementing this invention. This invention includes image memory 201 storing the image to be processed.

This image memory has a pixel organization such as illustrated in Figure 1. Image data processor system 200 includes one or more image processors 211 and 221. Each image processor 211 and 221 has a corresponding tile cache 213 and 223. The respective tile caches 213 and 223 are also connected to image processor system bus 205. Image processor system bus 205 is also connected to image memory 201 and may be connected to other image processor and tile cache combinations.

The primary advantage of using this technique of memory organization is reduction in the number and duration of accesses to image memory 210. This reduced memory traffic permits multiple processors, such as image processors 211 and 221, to work on image generation in parallel.

For the sake of comparison, assume that a typical page of text is approximately 10% dense, that is 1 in 10 display elements are part of the text strokes used to make the image.

Using the prior art memory organization, access to display elements in one direction of the two dimensional array can be accomplished within a DRAM row, page mode access. However,

display element access in the other direction must be random for images of any substantial size. Accesses within a DRAM row may be accomplished using page mode techniques which result in access times on the order of 50 nanoseconds per access whereas non-page mode accesses, page miss accesses, require access times on the order of 150 nanoseconds. According to this prior art memory organization, randomly accessing 10% of 256 display elements at a time would require about 25.6 accesses or 3840 nanoseconds for write only operations.

Using the memory organization of this invention, the memory accesses are not random but sequential. Thus page mode DRAM accesses may be used. Page mode DRAM accesses are on the order of 50 nanoseconds per access. To access 256 display elements in the tiled organization to load and writeback the tile cache requires 32 accesses, 16 reads and 16 writes. This requires only 1600 nanoseconds. This is a significant improvement over the 3840 nanoseconds required by the prior art memory organization. This invention requires 1600/3840 or 42% of the memory access time of conventional linear organized memory.

Figure 3 illustrates a block diagram of a TMS320C82 digital signal processor (DSP) in an image data processing system according to this invention. The tiled memory organization shown can be very efficiently implemented on a multiprocessor DSP such as the Texas Instruments TMS320C82. The basic architecture of this DSP is shown on Figure 3.

The multiprocessor DSP is a single integrated circuit 180. Integrated circuit 180 a fully programmable parallel processing platform that integrates two advanced DSP cores

DSP 181 and DSP 182, a reduced instruction set computer (RISC) master processor (MP) 183, multiple static random access memory (SRAM) blocks 185, 186 and 187, a crossbar switch 184 that interconnects all the internal processors and memories, and a transfer controller (TC) 188 that controls external communications. Transfer controller 188 is coupled to image memory 190 via bus 195. Note that transfer controller 188 controls all data transfer between integrated circuit 180 and image memory 190. Image data is stored in image memory 190 in tiles as illustrated in Figure 1.

In operation, the individual DSPs 181 and 182 operated independently on separate tiles. Each DSP 181 and 182 signals transfer controller 188 to transfer a tile of data from image memory 190 to the corresponding SRAM 185 and 186. The DSPs 181 and 182 perform a programmed image transformation function on the tile data in place in the corresponding SRAMs 185 and 186. Access by DSPs 181 and 182 and master processor 183 to SRAMs 185, 186 and 187 is mediated by crossbar switch 184. When complete, the DSPs 181 and 182 signal transfer controller 188 to transfer data back to image memory 190 for storage in the memory allocated to the corresponding tile. This cache-like technique greatly reduces the memory transfer requirements of image memory 190. Master processor 183 is preferably programmed for high level functions such as communication with other parts not shown.